2. Ba Tu

Program Name: BaTu.java Input File: batu.dat

Ba Tu has just learned to play Sudoku, but still needs help figuring out what values any particular square could be. The rules are that every row, column, and every 3X3 sub-box within the large grid have unique values from 1 to 9.

Your job is to help Ba Tu get started with any particular empty box, showing him what digit, or digits, could possibly work for that empty box.

In the puzzle shown here, the empty blank at position 1,1 at the top left corner of the puzzle cannot be a 5, 1, or 4 since that row already contains those numbers, and it cannot be 2, 3, or 9 since that column contains those, and it cannot be a 7 since there is a 7 in the sub-box for that blank. That means that 6 or 8 would work in that blank.

_ 5 _ 1 _ 7 _ _ _	_ 1 _ _ _ _ 9 _ 5	_ 4 _ 6 _ 2 _ _ _
2 _ 8 _ 4 _ 9 _ 1	_ 3 _ _ 7 _ _ 8 _	5 _ 1 _ 2 _ 4 _ 6
_ _ _ 3 _ 4 _ 2 _	4 _ 1 _ _ _ _ 6 _	_ _ _ 7 _ 9 _ 1 _

For the blank at row 4, column 2, between the 2 and 8, the only values that could work are 6 and 7, since that row already has the digits 1, 2, 3, 5 and 8, the column also has 4, and that sub-box also contains the 9.

Input: Nine strings of nine digits, each on a separate line, representing the beginning values of a 9X9 Sudoku puzzle. A zero digit represents an empty blank. After the initial nine strings will be several pairs of integers representing the column and row of an empty blank, each pair on a separate line.

Output: All the possible digits that could be placed in the empty blank indicated by each pair of digits. The digits are to be listed in ascending order with no spaces between.

Sample input:

```
050010040
107000602
000905000
208030501
040070020
901080406
000401000
304000709
020060010
1 1
4 2
2 5
3 5
```

Sample output:

```
68
67
4
24
```